UCRL-ID-120736

An Implementation of a Medium Resolution Minefield Model in the Joint Conflict Model

Jeffrey E. Pimper
JoAnn Matone

January 13, 1995

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

An Implementation of a Medium Resolution Minefield Model in the Joint Conflict Model

Jeffrey E. Pimper
JoAnn Matone

January 13, 1995

This document was prepared as an account of work sponsored by the United States Marine Corps Combat Development Center. Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. This work is not to be reproduced without the express permission of the authors.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

ABSTRACT

An implementation of a new, flexible, and realistic representation of conventional minefields in the Joint Conflict Model (JCM) is presented. The model includes important aspects of minefield effects on battlefield entities and of breaching devices on minefields. The model is designed at "medium resolution," that is, it is general enough to depict a wide variety of tactical situations accurately; however, it only represents tactically significant aspects of mine warfare, discarding or aggregating details, thus minimizing computer memory and speed requirements. This paper describes the model in detail, its implementation in the JCM simulation code, and its use in a preliminary analysis effort related to the effect of delay on the tactical battlefield.

1.0 MODEL DESCRIPTION AND ALGORITHMS

1.1 Review of Minefield Model.[1] Conventional mines are not, in general, used in isolation, but are combined with other counter-mobility techniques into a complex obstacle. Using the new minefield model, the representation of complex obstacles becomes possible in JCM simulations.

The model is designed in an object-oriented fashion (shown in Figure 1). A minefield consists primarily of a list of mine regions, but also includes an enclosing polygon and rectangle, used for quick determination of whether an entity will encounter a minefield in any given step. The mine region is designed to be an area with uniform properties (such as mine density). The mine densities can be thought of as a series of layers of mines, one on top of the other (see Figure 2). A minefield's regions change dynamically in shape and number as entities interact with them.

Associated with each mine region is:

a convex polygonal shape,
a series of mine types,
a series of corresponding mine densities,
a list of individually placed mines.

The mine type is an index into a table of different types of mines. Associated with each mine type is:

a name,
a count of the number of times this type of mine must be triggered before exploding.

Individually placed mines occur only under certain circumstances (see the discussion of indeterminate placement below for a detailed

[1] UCRL-ID-117763; Edward T. Powell, A Medium Resolution Minefield Model Suitable for Entity-Level Resolution Combat Simulations; Lawrence Livermore National Laboratory, June 9, 1994.

description of these circumstances.) An individually placed mine has associated with it:

a mine type,
a count of the number of times this particular mine must be triggered before exploding,
an explicit position on the ground.

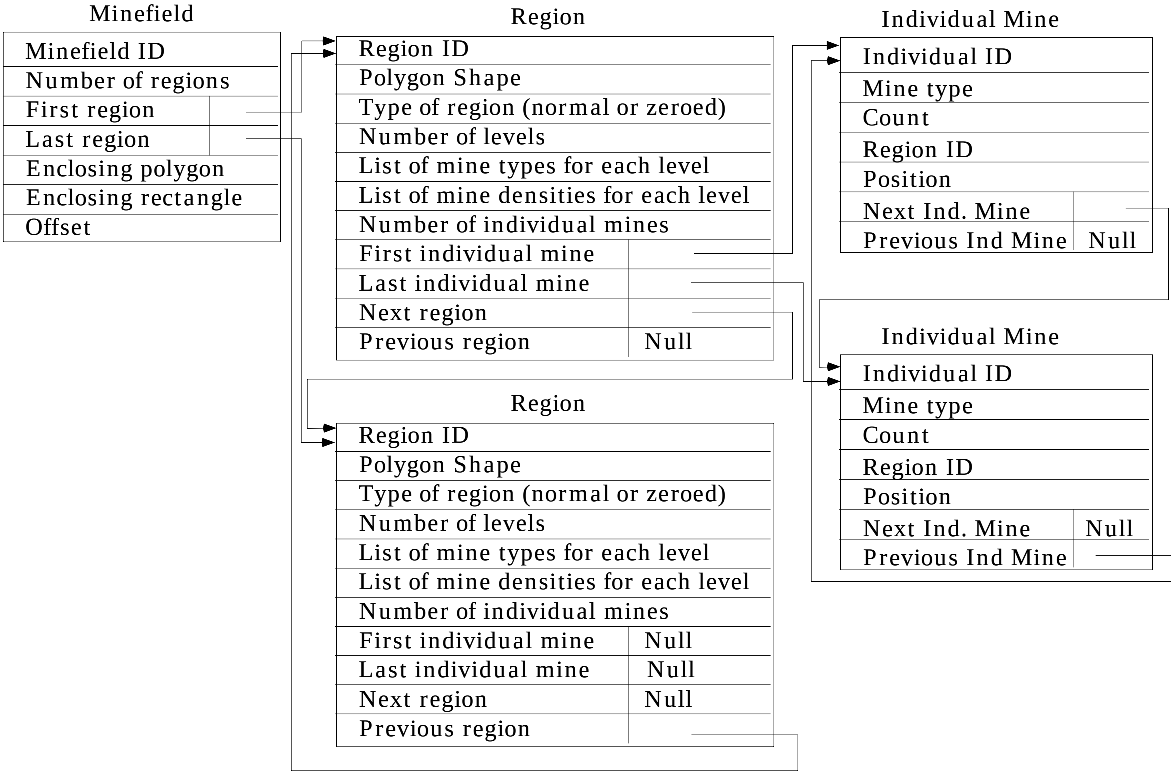

Figure 1: Minefield model data description

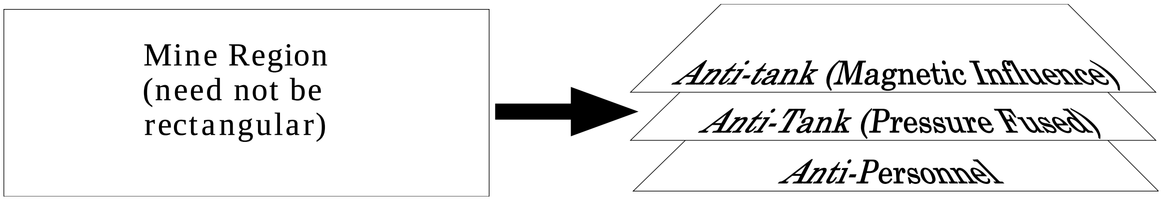

Figure 2: Minefield model layer concept

Because entities interact with mines, certain entity parameters must be defined to make the minefield model workable. The primary attribute of an entity in this context is its platform type. The platform represents the physical description of the entity on the battlefield. With the new model, it is possible for an entity to have two different sets of platform characteristics. Two sets are necessary to distinguish the mobility and vulnerability characteristics of the platform while in normal operation and also while breaching. For instance, a tank with a plow attached has very different effects on a minefield depending on whether the plow is up or down. Each system type has a single set of characteristics which represents its own platform while it is breaching (i.e. a tank with plow down). Each mine type has nine sets of characteristics, one for each generic platform class, which represent all platforms while not breaching (i.e. plow up). An entity with no breaching capability uses only the mine type's generic set of platform characteristics for its own platform class. The system attributes that are needed for interaction with a minefield are: the vehicle's length and width, its speed, and a description of its areas of effect. The areas of effect are illustrated in Figure 3. A system may have up to seven areas of effect depending on how detailed a model the analyst requires. A tank could be modeled with just one area of effect, representing an average over the whole vehicle. Or it could have three areas, representing the two treads and the underside. This formulation gives the analyst maximum flexibility for defining systems in as much detail as needed, without having to include all of the detail all of the time.

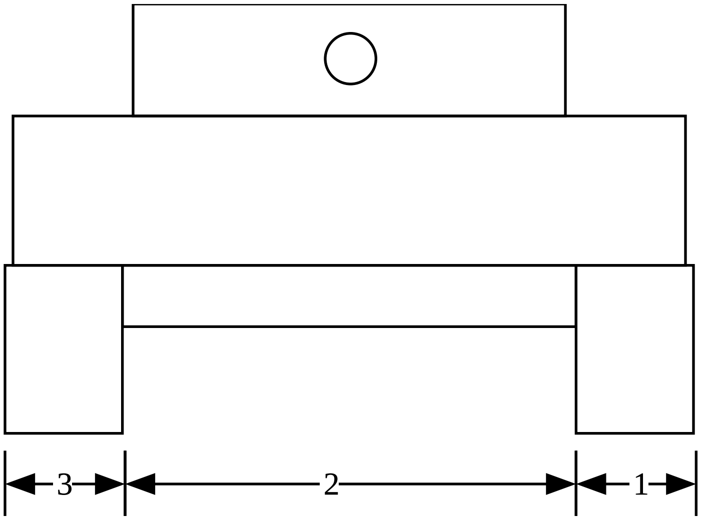

Figure 3: An entity's areas of effect, numbered from the entity's left.

Note that the area of effect represents the effect of the *system* on the *minefield*, not vice versa. Areas of effect are intended as a means of incorporating the different types of breaching mechanisms, not to represent areas of differing vulnerability of the system to the mines. This feature was designed for computational efficiency, and is (in the context of the current simulation) an adequate approximation to reality, because JCM makes no distinction between mobility kills, firepower kills, and catastrophic kills. This feature, however, could be altered in the future if needed.

To describe the interaction of entities and mines, three pairs of tables are needed. One table of each pair is defined by the system type and is used by systems which are currently breaching. The other table of each pair is defined by the mine type and is used for systems which either do not have any breaching capability at all or have their breaching capability currently turned off.

One table pair contains the probability of a mine being destroyed (or otherwise rendered non-functional) by the entity when encountered. The system type table of the pair has an entry for each mine type and each area of effect for that system, and represents the probability that a mine of that mine type will be destroyed when encountering that particular area of effect on that particular system. The mine type table of the pair has an entry for each platform class, and represents the probability that a mine of that mine type will be destroyed when encountering a system having a platform belonging to that particular platform class. A similar pair of tables is required for the probability of the mine being triggered. The final pair of tables represents the probability that a mine will kill a system.

In the JCM simulation, entities move in discrete steps, the size of the move depending on whether the entity can maintain its desired speed over the terrain. Though the size of the step can be set by the user to any value desired, a default size of approximately 50 meters is generally used. As the entity steps from one position to the next, it checks to make sure there are no obstacles that would prevent its movement. The new minefield model relies on this step-wise movement model.

The basic idea behind the algorithm for entity/minefield interaction is shown in Figure 4. (a) An entity approaches a minefield. (b)-(c) It is determined that the entity's path will intersect the minefield in the current step, and an assessment of the minefield's effect on the entity must be made to determine the actual size of the step. (d) Once this determination is made (as well as the assessment of whether the entity was killed or not), and the extent of the entity's movement is known, the effect the entity had on the minefield can be determined. It is very important to note that these two assessments are separated. First, each minefield's potential effect on the entity is determined. If the minefield "kills" the entity, then the entity's movement step is reduced to the point of the "kill" and the new step is used when evaluating further minefields. After all minefields have been evaluated, and the entity's final step size is determined, then the entity's effect on the minefields is determined. If these two assessments were not separated, it would be very difficult to efficiently account for overlapping minefields.

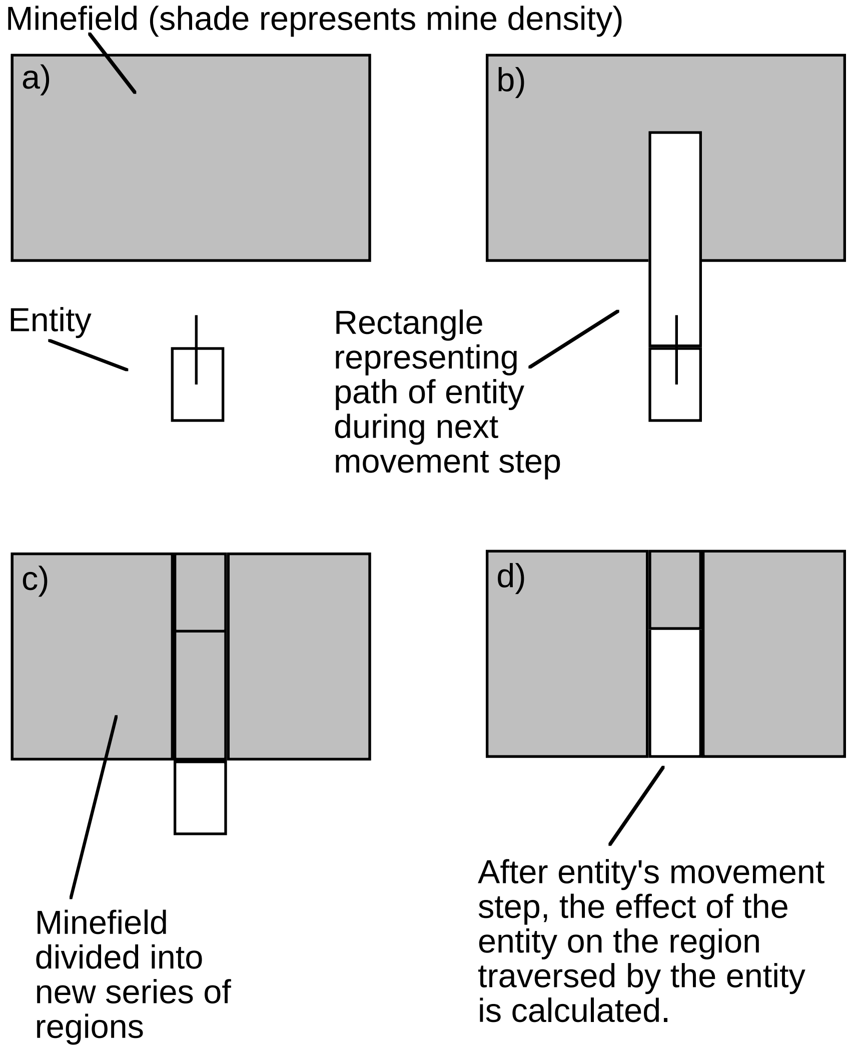
Minefield (shade represents mine density)
a)
b)
Entity
Rectangle representing path of entity during next movement step
c)
d)
Minefield divided into new series of regions
After entity's movement step, the effect of the entity on the region traversed by the entity is calculated.

Figure 4: Interaction of an entity with a minefield. a) The entity approaches a minefield. b) A rectangle representing the entity's next movement step is calculated. c) The entity path rectangle is intersected with the minefield, yielding a number of regions. d) After the effect of the minefield on the entity is calculated, the effect of the entity on the

minefield is determined, and the minefield regions are updated accordingly.

Regions traversed by the entity are divided into a number of smaller regions: regions outside the entity's areas of effect (which are unchanged in content), and regions that the entity interacted with (which can be changed in content). Regions that interact with the entity can change in two ways. First, if a region contained individual mines, these mines could have exploded, and thus need to be removed from the region. Second, if the region contained mine density, this density could be altered by the entity's passage through the region. As more and more entities or breaching devices interact with the minefield, more and more regions are created. A technique for limiting the proliferation of regions is needed to prevent an exponential explosion in the number of regions. Such a technique is described below.

Mines are only given positions on the ground when needed. Mine densities are used whenever possible. This is the concept of indeterminate placement. When assessing the effect that traversing a mine region has on an entity, that region's mine density is temporarily transformed (i.e. deaggregated) into a list of individually placed mines positioned randomly in the region. The effects of these mines are determined in order of encounter. If any of the mines kill the entity, the entity is said to have died at the position of that mine, the assessment process is stopped, and the length of the entity's movement step is recorded. The sequence of events is summarized in Figure 5 and is expanded upon in the Appendix. Note that the minefield has not been changed in any way so far. Since it is possible for an entity to encounter more than one minefield in any one movement step, it is necessary to repeat the maximum movement step calculation for each of the minefields that the entity encounters in a step. Then the smallest maximum movement step achieved is said to be the actual movement step.

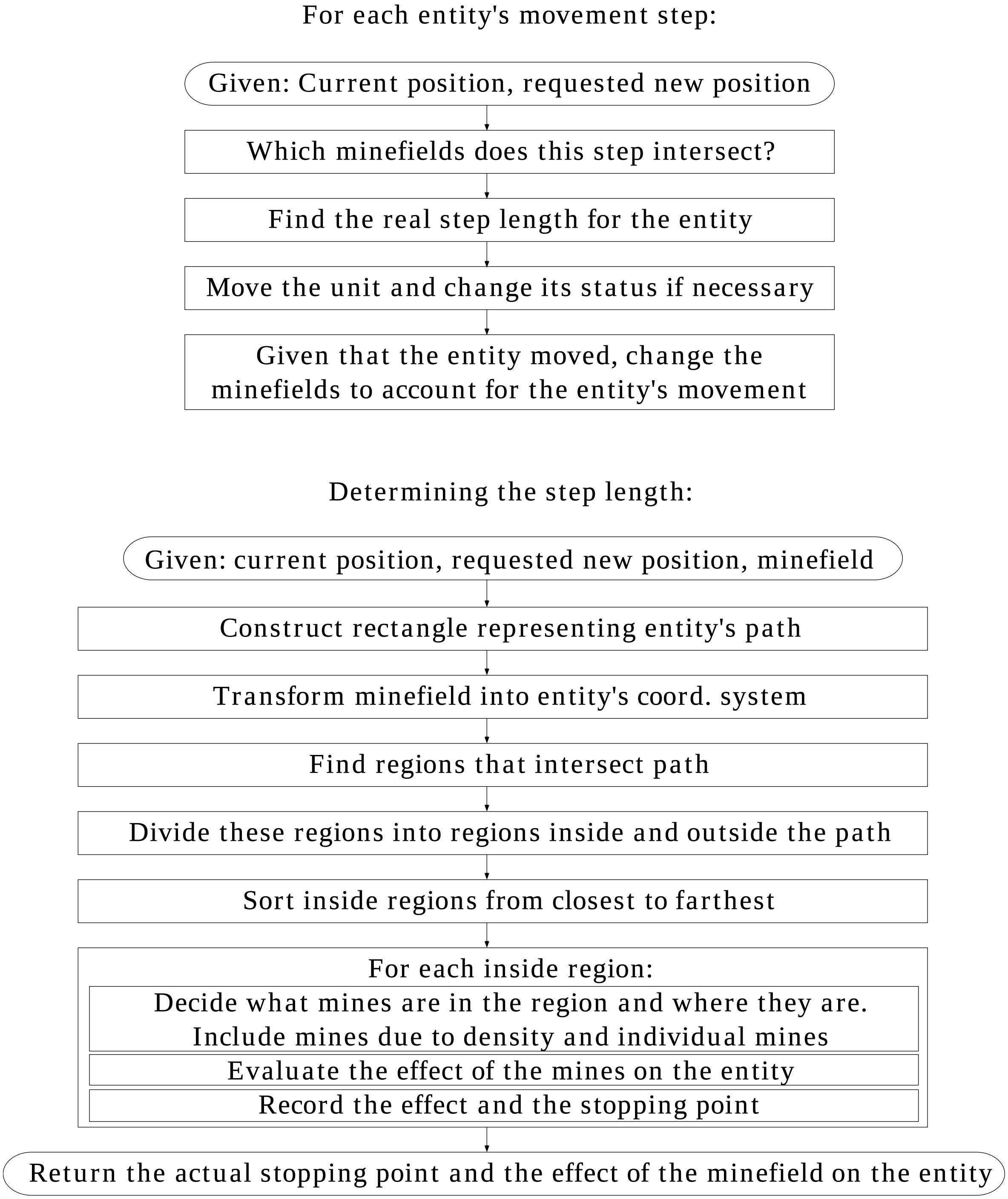
For each entity's movement step:
Given: Current position, requested new position
Which minefields does this step intersect?
Find the real step length for the entity
Move the unit and change its status if necessary
Given that the entity moved, change the minefields to account for the entity's movement
Determining the step length:
Given: current position, requested new position, minefield
Construct rectangle representing entity's path
Transform minefield into entity's coord. system
Find regions that intersect path
Divide these regions into regions inside and outside the path
Sort inside regions from closest to farthest
For each inside region:
Decide what mines are in the region and where they are. Include mines due to density and individual mines
Evaluate the effect of the mines on the entity
Record the effect and the stopping point
Return the actual stopping point and the effect of the minefield on the entity

Figure 5: Movement and step length logic.

Once the actual movement step is calculated, it is now possible to assess the effect of the entity on the minefield. Each minefield is updated using the knowledge that the entity moved as far as it did. That is, given that the entity moved a certain distance (before either being killed or reaching the end of its step), each mine that the entity encountered during the step must have either: (1) been unaffected by the entity (i.e. an anti-armor mine encountered by a dismounted troop), (2) destroyed (or otherwise removed) by the entity, (3) counted down but not triggered, (4) triggered, but did not kill the entity, or (5) triggered and did kill the entity. Though cases (2) , (4), and (5) have similar results, (i.e. the mine is gone), the difference is that in cases (4) and (5) the mine explodes (and can thus be detected), whereas in case (2) the mine is destroyed without possibility of detection. Mine densities are adjusted using the table values for the probabilities of each of these possibilities. Individual mines are left in the minefield or removed depending on random numbers drawn against each of the same probabilities.

The use of individual mines in mine regions is an important aspect of the medium resolution model. Individual mines can appear in three basic situations. First, individual mines associated with a minefield region can be defined at the beginning of a scenario. Therefore, it is possible for the user to specify in advance the types and positions of individual mines. Second, if a mine designed to explode only after being triggered more than once is encountered in the assessment calculation while density mines are deaggregated, that mine is not reaggregated into the mine density, but is left in the region as an individual mine. The trigger count of the mine is decreased by one and the mine density (after any other mines are reaggregated) reflects the lack of that mine which is left as an individual.

Finally, a region with a low density of mines is automatically deaggregated into a region with no mine density, just individual mines.

It is this last rule, combined with two other rules, that prevents the number of regions from increasing exponentially as more and more entities interact with a minefield. These rules can be stated as follows: (1) If the expected number of mines in a region (that is the mine density times the region's area), becomes less than a fixed number *N*, the representation of that region is changed from mine density to individually placed mines. Such a region is said to be *zeroed*, in that the density has been zeroed out in favor of an individual mine representation. (2) Zeroed regions are never subdivided any further. This rule places an upper limit on the number of regions into which a minefield can be divided. (3) Regions that have neither mine density nor individual mines are deleted from the minefield. This rule tends to decrease the number of regions in a minefield. In practice, when these rules are implemented, the number of regions in a typical minefield seldom rises above twenty.

Finally, it is useful to examine the algorithm used to determine the intersection of an arbitrary polygon with the entity path rectangle. This algorithm is the basis for dividing up the minefield into regions traversed by a vehicle and regions not encountered. The algorithm, illustrated in Figure 6, is optimized for the path being rectangular. First, any region which intersects the path rectangle is transformed into the coordinate system in which the path rectangle is horizontal with the entity traveling "east." Second, it is "sliced," first to the left of the path, then to the right. Finally, the region is sliced by the trailing edge of the path, and then by the leading edge. Slicing the region in this fashion is very efficient, since a general intersection algorithm need not be developed. Also, since the region is first sliced to the sides of the entity's motion, the number of subsequent polygons encountered in the next step is minimized. (Note that in Figure 4 (d), the entity, if it survives, will only traverse one additional region.)

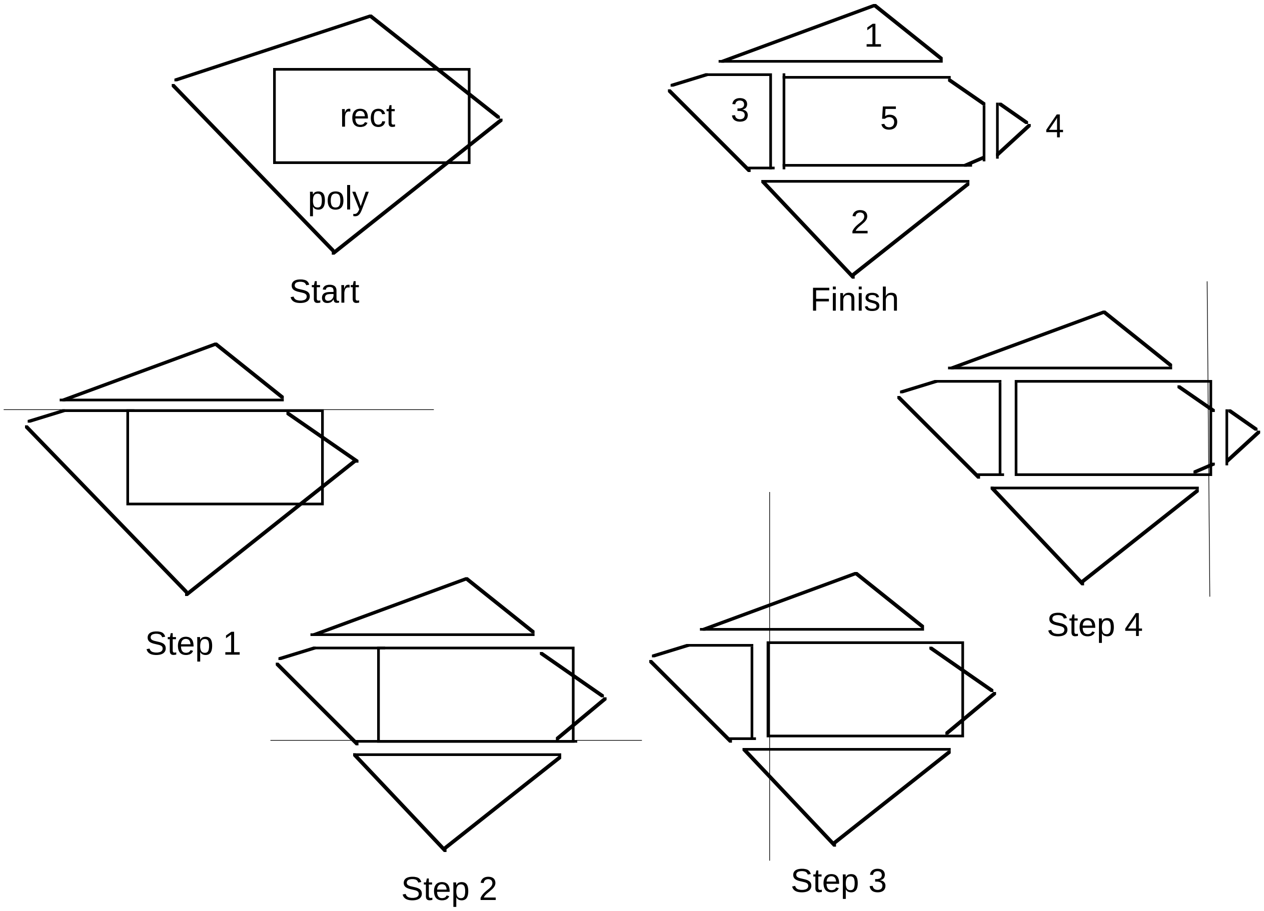

Figure 6: Polygon splitting algorithm

In summary, the new minefield model allows the analyst maximum flexibility to define a wide range of different types of minefields and breaching devices. A minefield can be designed with all individual mines laid out in any manner desired. Or a minefield can be defined as a list of regions, each region containing a series of mine types and densities. In the latter case, the mines are laid down only when necessary to evaluate their effects on a passing entity, or the effect of the entity on the mines. Regions with mine density are used unless the expected number of mines in a region is small, in which case those regions are zeroed to eliminate mine density, and represented as indivisible regions with only individual mines.

1.2 JCM Integration Issues. The minefield model was designed, tested, and implemented in a testbed simulation. This development method was chosen so that the model could be developed independent of any particular simulation, and in principle could be inserted into any entity-level simulation. For the purposes of evaluation and analysis, the minefield model was implemented in JCM. Because this code was the first one to use the new minefield model, it is important to list the issues raised by incorporating the model into this specific simulation.

JCM is supported with various specialized data editors and post-run analysis tools. Graphics terminals are used to display the terrain and the simulated activities such as movement and engagement. Integrating the new minefield model involved making changes in the movement algorithm, the display of simulated activities, and the data input and output. The integration effort had two constraints: It had to be done in a reasonable time period, and the new code had to be able to read all of the same input files that the unmodified code could read.

In JCM's previous minefield model, breaching was modeled as an activity accomplished over a period of time without movement. In the new model, an entity can move some distance into the minefield without setting off a mine, even if it has no breaching capability. To accommodate the new minefield model, additional data structures and movement logic were developed. The additional data structures contain all of the necessary minefield data shown in Figure 1. The added movement logic is only used to model movement through minefields. The original minefield model is bypassed.

The new minefield model as originally designed and implemented was predicated on the fact that each entity in the simulation would represent a single system. However, JCM allows units or entities to be defined that contain more than one item system. When units move, they normally move in a column formation. If a unit is in the assault, it moves with its item systems in a line formation. Moving each individual item system separately would be too costly, in terms of computation time, if there were a large number of item systems, so a trade-off was implemented that preserves unit integrity yet allows each item system to interact with a minefield as an individual.

If a unit is moving in a line formation (assaulting), then each item system is moved separately and interacts with each minefield as an individual. Fortunately, this case is rare so the extra cost of calculating each item system's individual movement step can be borne. x* In JCM, normally units move in a column formation. When a unit in column formation encounters a minefield the unit is treated as if all of its item systems occupy the same point and take the same movement step. Each item system takes its step individually, in formation order. Hence if a mine must be triggered three times before it detonates, then the first two systems in the formation that trigger the mine pass over it with the only

* It is rarer in real life. A tactical unit finding itself in a situation like this one would not continue to blindly move forward. At the very least, it would return to a column formation to “bull” its way through the minefield.

effect being the mine counts down by one for each system. When the next system in the formation triggers the mine, its count goes to zero and the mine detonates. Whenever a system encounters a mine, the probability of triggering the mine is determined from the Mine Type Characteristics data if the system is not breaching and from the system's Breach Lane Probabilities if the system is breaching.

With the original JCM minefield model, an entity had two options when it encountered an unbreached minefield: either breach the minefield if it was capable of breaching, or halt. With the new minefield model, if the entity decides (or is ordered) to "bull" through a minefield it must be able to move through the minefield without being in breach mode. Therefore, JCM now allows the player to issue the following movement commands to the entities under his control: unconditionally move in breach mode (breach on), move in breach mode as required (auto breach), and do not use breach mode while moving (breach off). To implement these commands the player chooses options from the JCM Planning and Movement menus (Figures 7 and 8) and then picks the entity for which the option applies. To set a unit’s breach mode, the player selects "Breach" from the Planning menu and picks the unit with the appropriate puck button. Breach "on" is selected by picking the unit with the white or green buttons. Breach "auto" is selected by picking the unit with the yellow button and breach "off" is set by using the blue button.

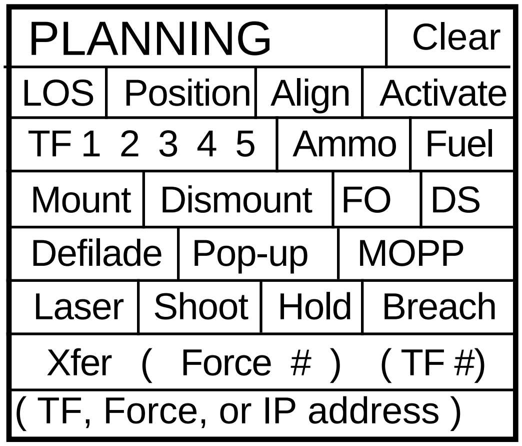

Figure 7: The JCM Planning Plan

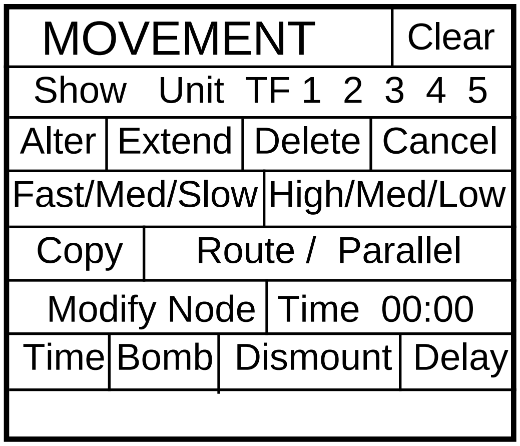

Figure 8: The JCM Movement menu.

Units in JCM can be ordered to proceed at any of three movement rates, called "Slow," "Medium," and "Fast." These speeds are a characteristic of each node in a unit’s movement path and are selected by picking the desired speed from the JCM Movement menu when creating the path. The actual speeds represented by "Slow," "Medium," and "Fast" are unit type characteristic data and are specified by the analyst using the JCM scenario editor. Units moving with breach on move at their breach speed regardless of the movement node speed. Units with breach set to auto move at their movement node speed until they detect a minefield then they move at their breach speed.

The effect of a minefield on the movement of units that are placed in the "no" breach mode depends on the unit's current movement status, illustrated in Figure 9. If a unit is in "no" breach mode and it detects mines (see below) then its reaction to the mines depends on the movement node's speed. If the unit's movement rate is "Slow," then the unit will stop upon detection, convert its current movement node to a "Stop" node, and a message will be displayed on the controlling player's workstation informing him of what happened to the unit. If the unit's movement rate is either "Medium" or "Fast" then the unit will not stop. Instead it will attempt to "bull" its way through the minefield at its medium or fast speed.

Units not breaching will move at their "Assault" speed if they are firing direct fire weapons at targets at ranges less than their "Assault Range." The "Assault Range" and "Assault Speed" are JCM unit characteristic parameters which are defined in the JCM Scenario Editor. If an assaulting unit is also breaching it moves at its breach speed.

Minefield Status	Breach Mode	Speed				
		Assault	Slow	Medium	Fast	Breach
not detected	Off	on-line	column	column	column	----
	On	----	----	----	----	column
	Auto	on-line	column	column	column	----
not detected but detonated	Off	on-line	stop	column	column	----
	On	----	----	----	----	column
	Auto	on-line	breach	breach	breach	----
detected	Off	on-line	stop	column	column	----
	On	----	----	----	----	column
	Auto	on-line	breach	breach	breach	----

Figure 9: Minefield Action State Table

Minefields are not acquired as individual minefields or minefield regions. Rather, units detect that they are in the presence of mines. When a unit detects any type of minefield either by detonation or acquisition, the unit is highlighted on its controlling player's screen by having a magenta colored circle drawn around it. The magenta circle moves with the unit as long as the minefield continues to be acquired or until the time since the last acquisition exceeds the unit's "Time on Line." No information about the size or shape of the minefield or its regions is reported to the player. Because acquisition is based on a distance from a mine, a unit may not actually be in a minefield when acquisition occurs. A player can put an entity into breach mode (for its own protection) and then scout around the area to determine the probable extent of the minefield.

When mines are acquired by an entity in auto-breach mode, the entity immediately goes into breach mode.

Acquisition of mines in a minefield is based on the distance between the entity and the minefield region containing the mines, illustrated in Figure 10. The value P_1 represents the probability that an entity will realize that it is in the presence of mines when it actually is within a minefield region. The value P_2 represents the probability that it detects the mines in a region at a distance equal to R_{max}, the maximum acquisition range. For distances between zero and R_{max}, the corresponding probability is a linear interpolation between P_1 and P_2. For distances greater than R_{max}, an entity will not acquire the minefield. At each movement step while an entity is near a minefield, a random draw is compared to the probability of acquisition of the mines in each region of the minefield. If the random draw is less than the probability, the entity acquires the mines in the region; otherwise, it does not. The values of P_1 and P_2 are assigned by type of mine and apply to all minefields equally, so that each minefield inherits its acquisition probabilities from the mines that comprise it.

There were some changes made in JCM to the graphical display of the interaction between minefields and entities. In the old model, every explosion caused a kill. In the new model, mines can explode without causing a kill. A "mine explosion" graphic is now used to designate

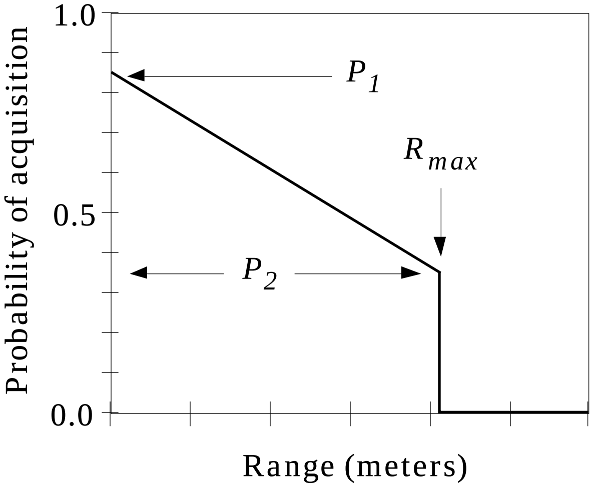

Figure 10: Simple model for the acquisition of minefields.

when a mine detonates. When a mine kill occurs, it is designated with the usual JCM symbol for a mine kill: "V".

The improved minefield model required additional data to function with JCM. All of the minefield-specific data is currently stored in a file separate from the regular JCM data files. This file is named SCENARnn.MINE, where "nn" is the same number as the scenario's force organization file, SCENARnn.ORG. The JCM Scenario Editor was modified to allow the user to modify the data in the minefield file (if present). Though the new minefield model was designed to be used with any entity-level simulation, it can be seen that a number of issues (such as graphics representation and data editing capabilities) need to be worked through to arrive at a truly integrated model.

To summarize, a need existed for a minefield model that retains enough fidelity to perform tactically meaningful analyses, but did not clutter the simulation with a lot of unnecessary detail: a medium resolution model. A model that fits these criteria has been developed independent of any single entity-level combat simulation. In this model, minefields are represented as polygonal regions of mine density, and individual mines are positioned only when necessary in a technique known as indeterminate placement. Efficient algorithms have been developed to manipulate these regions.

APPENDIX: MINE MODEL LOGIC

A.1 THE EFFECT OF A MINE REGION ON A SYSTEM

```
    Input: (region, system type, rectangle representing entity's path)
    Output: (list of mines and their potential effects)

Place all mines in the region into a mine list, sorted by distance from the
  system

For each mine in the mine list do:
Begin Loop
    Get mine type

    IF mine is a density mine
        Randomly place the mine in the region

    IF entity is breaching
        Find which area of effect this mine is in

        Get Pmd = prob_mine_being_destroyed(area of effect, system type,
          mine type)  [The probability that the system will destroy this type
          of mine when the mine is encountered in a particular area of
          effect.]

        Get Pmt = prob_mine_trigger(area of effect, system type, mine type)
          [The probability that this mine type will trigger when exposed to
          this system.]

        Get Pmk = mine probability of kill(area of effect, system type, mine
          type)

    ELSE
        Get Pmd = prob_mine_being_destroyed(platform class, mine type)
          [The probability that the platform type  will destroy this type of
          mine.]

        Get Pmt = prob_mine_trigger(platform class, mine type)  [The
          probability that this mine type will trigger when exposed to this
          platform class.]

        Get Pmk = mine probability of kill(platform class, mine type)

    Get RND [a random number between 0 and 1]

    IF $RND \le Pmd$
        Mark mine as potentially destroyed
        Next mine
    ELSE IF RND > Pmd
        (Mine is not destroyed)

        Get new RND
        IF RND > Pmt
            (Mine does not trigger)

            Delete mine from list
            Next mine

        ELSE
            (Mine triggers)

            IF mine count > 1
                Mark mine as potentially counted
                Next mine

            ELSE
                (Mine detonates)

                Mark mine as potentially detonating

                Get new RND
                IF RND > Pmk
                    (No kill)

                    Next mine

                ELSE
                    (Mine kills)

                    Mark mine as potential killer
                    Delete all following mines from the list
                    Exit loop
End Loop
END of assessment of effect of minefield on system
```

A.2 THE EFFECT OF A SYSTEM ON MINE REGIONS

Input: (list of mines and their potential effects, system type)
Output: (effect on system, changed regions)

The mine list contains all mines in all regions which have been potentially affected by the system

Sort mine list based on mine distance from system's starting position

For each mine in the mine list do:

```
Begin Loop
    IF potential effect is "No Effect"
        Next mine
    ELSE IF potential effect is "Mine Destroyed"
        IF mine was a density mine
            Reduce the density of the region to which it belongs by one mine
        ELSE
            Delete mine from region's list of individual mines

    ELSE IF potential effect is "Mine Counted"
        IF mine was a density mine
            Convert it to an individual mine
            Reduce the density of the region to which it belongs by one mine
            Add mine to region's list of individual mines

        Decrement mine's count by one

    ELSE IF potential effect is "Mine Detonated"
        IF mine was a density mine
            Reduce the density of the region to which it belongs by one mine
        ELSE
            Delete individual mine from region's list of mines

        Display mine detonation graphic on user workstations

    ELSE (potential effect is "Mine Killed")
        IF mine was a density mine
            Reduce the density of the region to which it belongs by one mine
        ELSE
            Delete individual mine from region's list of mines

        Display mine detonation graphic on user workstations

        Kill the system and inform its owner

        Exit loop (ignoring any further mines on the list)
End Loop

END   of assessment of effect of system on minefield
```

ACKNOWLEDGMENTS

The authors wish to thank Arnold Warshawsky, Stephen Wong, and Michael Uzelac for their assistance in performing this research.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551